

Esercizio 1

Si scriva un programma in linguaggio C++, a linea di comando, che accetti la seguente sintassi:

```
frequenze <input file> <output file>
```

Il programma prende in input un file binario e per ogni byte ne conta le occorrenze. In output produce un file di testo composto da una riga per ogni byte presente nel file di input con il seguente formato:

```
<byte><tab><occorrenze><a capo>
```

Il byte è rappresentato con il suo valore esadecimale a due cifre, le occorrenze in base dieci.

Infine invia su standard output l'entropia dell'insieme di simboli secondo la distribuzione appena calcolata, secondo il formato:

```
Entropia: <valore><a capo>
```

Il valore è espresso come numero decimale.

Esercizio 2

Si scriva un programma in linguaggio C++, a linea di comando, che accetti la seguente sintassi:

```
packbits c <input file> <output file>
```

```
packbits d <input file> <output file>
```

Quando viene specificata l'opzione "c" il programma apre il file specificato (il file deve essere trattato come file binario, ovvero può contenere un valore qualsiasi da 0 a 255 in ogni byte), lo comprime con l'algoritmo Packbits e lo salva in un nuovo file (utilizzate per chiarezza l'estensione ".pkb"). Quando viene specificata l'opzione "d", il programma tenta di decomprimere il contenuto del file di input salvandolo nel file di output.

Se l'input fosse:

```
aaaaabbbbcbdefghaaaaaaaaaaaa
```

il file di output diventerebbe (ogni casella è un byte):

252	a	252	b	5	c	d	e	f	g	h	246	a	128
-----	---	-----	---	---	---	---	---	---	---	---	-----	---	-----

e visto in un editor esadecimale sarebbe:

```
FC 61 FC 62 05 63 64 65 66 67 68 F6 61 80
```

Nota: Inizialmente si tratti ogni ripetizione come una run. Come ulteriore estensione, si realizzi il programma che nel caso ci sia una copia di byte, seguita da una run da 2 byte, ancora seguita da una copia di byte, non generi la run, ma la incorpori in un'unica copia. Ad esempio, si consideri la sequenza:

```
xyaabcd
```

Compressa con l'algoritmo base, diventerebbe:

1	x	y	255	a	2	b	c	d	128
---	---	---	-----	---	---	---	---	---	-----

Con la variante considerata invece:

6	x	y	a	a	b	c	d	128
---	---	---	---	---	---	---	---	-----

Portando ad un risparmio di 1 byte.

In generale la soluzione non deve caricare l'intero file in memoria o tenere l'intera sequenza compressa o decompressa in memoria. Tutto deve essere fatto riducendo il più possibile la quantità di dati tenuta in RAM.